

CLAIMS

1 1. A method of performing data classification wherein data having a data header that
2 is classified according to applicable rules, the method comprising the steps of:

3 a) creating an empty bitmap including a number of bits equal to a number of ap-
4 plicable rules;

5 b) applying the applicable rules to the data header and for each rule that matches
6 the data header,

7 c) setting a bit corresponding to the rule in the bitmap; and

8 d) storing the bitmap in an entry that corresponds to the data header in a lookup
9 table.

1 2. The method as in claim 1, further comprising the steps of:

2 e) assigning an index number to the bitmap; and

3 f) storing the index number as an index to the entry that contains the bitmap cor-
4 responding to the data header, wherein the index number is associated with
5 the data header.

1 3. The method as in claim 2, further comprising the steps of:

2 iterating the data header from zero to a maximum value;

3 and for each iterated value of the data header,

4 performing the steps of a) – f).

1 4. The method as in claim 3, further comprising the steps of:

for each bitmap corresponding to the iterated value of the data header;
determining if there is a matching bitmap previously created;
assigning a new index number as an index to a new entry that contains the bitmap
in the lookup table if there is no previous bitmap that matches; and
assigning the index number of the previous bitmap to the bitmap if the bitmap
matches the previous bitmap.

5. A method of performing data classification wherein data having a data header that
is classified according to applicable rules, the method comprising the steps of:
a) dividing the data header into a plurality of fields; and for each field,
b) creating an empty bitmap including a number of bits equal to a number of ap-
plicable rules;
c) applying the applicable rules to the field and for each rule that matches the
field,
d) setting a bit corresponding to the rule in the bitmap; and
e) storing the bitmap as an entry that corresponds to the field in a lookup table
associated with the field.

6. The method as in claim 5, further comprising the steps of:
f) assigning an index number to the bitmap; and
g) storing the index number as an index to the entry that contains the bitmap cor-
responding to the data header, wherein the index number is associated with
the data header.

7. The method as in claim 6, further comprising the steps of:

iterating each field from zero to a maximum value;

and for each iterated value of the field,

performing the steps of b) – g).

8. The method as in claim 7, further comprising the steps of:

for each bitmap corresponding to the iterated value of the field;

determining if there is a matching bitmap previously created;

assigning a new index number as an index to a new entry that contains the bitmap

in the lookup table if no previous bitmap matches; and

assigning the index number of the previous bitmap to the bitmap if the bitmap

matches the previous bitmap.

9. The method as in claim 8, further comprising the steps of:

cross-producting a plurality of lookup tables corresponding to a plurality of fields

of the data header to form a next level lookup table.

10. The method as in claim 9, wherein the step of cross-producting includes:

creating a next level bitmap from the cross-product of the plurality of bitmaps in

the respective lookup tables;

determining if there is a matching next level bitmap previously created from the

cross-product;

6 assigning a new next level index number as an index to a new entry that contains
7 the next level bitmap in the next level lookup table if no previous next level
8 bitmap matches; and
9 assigning the next level index number of the previous next level bitmap to the
10 next level bitmap if the next level bitmap matches the previous next level
11 bitmap.

1 11. The method as in claim 10, further comprising the steps of:
2 for N number of lookup tables, wherein N is an integer,
3 cross-producting the N number of lookup tables; and
4 repeating the cross-producting on resulting lookup tables until one final lookup
5 table remains.

1 12. The method as in claim 11, wherein the final lookup table contains bitmaps, each
2 bitmap representing a complete set of rules that match a data header value, the
3 method further comprising the step of using the bitmap to find the first matching
4 rule of the header value.

1 13. The method as in claim 11, wherein the final lookup table contains bitmaps, each
2 bitmap representing a complete set of rules that match a data header value, the
3 method further comprising the step of using the bitmap to find the best matching
4 rule of the header value.

1 14. The method as in claim 11, wherein the final lookup table contains bitmaps, each
2 bitmap representing a complete set of rules that match a data header value, the
3 method further comprising the step of using the bitmap to find all the rules appli-
4 cable to the header value.

1 15. A processor executable medium having instructions contained therein, which
2 when executed by a processor causes the processor to perform a method of per-
3 forming data classification wherein data having a data header that is classified ac-
4 cording to applicable rules, the method comprising the steps of:

- 5 a) creating an empty bitmap including a number of bits equal to a number of ap-
6 plicable rules;
- 7 b) applying the applicable rules to the data header and for each rule that matches
8 the data header,
- 9 c) setting a bit corresponding to the rule in the bitmap; and
- 10 h) storing the bitmap in an entry that corresponds to the data header in a lookup
11 table.

1 16. The processor executable medium as in claim 15, further comprising the steps of:
2 e) assigning an index number to the bitmap; and
3 f) storing the index number as an index to the entry that contains the bitmap cor-
4 responding to the data header, wherein the index number is associated with
5 the data header.

1 17. The processor executable medium as in claim 16, further comprising the steps of:
2 iterating the data header from zero to a maximum value;
3 and for each iterated value of the data header,
4 performing the steps of a) – f).

1 18. The processor executable medium as in claim 17, further comprising the steps of:
2 for each bitmap corresponding to the iterated value of the data header;
3 determining if there is a matching bitmap previously created;
4 assigning a new index number as an index to a new entry that contains the bitmap
5 in the lookup table if there is no previous bitmap that matches; and
6 assigning the index number of the previous bitmap to the bitmap if the bitmap
7 matches the previous bitmap.

1 19. A processor executable medium of performing data classification wherein data
2 having a data header that is classified according to applicable rules, the method
3 comprising the steps of:
4 a) dividing the data header into a plurality of fields; and for each field,
5 b) creating an empty bitmap including a number of bits equal to a number of ap-
6 plicable rules;
7 c) applying the applicable rules to the field and for each rule that matches the
8 field,
9 d) setting a bit corresponding to the rule in the bitmap; and

i) storing the bitmap as an entry that corresponds to the field in a lookup table
associated with the field.

20. The processor executable medium as in claim 19, further comprising the steps of:

j) assigning an index number to the bitmap; and
k) storing the index number as an index to the entry that contains the bitmap corresponding to the data header, wherein the index number is associated with the data header.

21. The processor executable medium as in claim 20, further comprising the steps of:

iterating each field from zero to a maximum value;
and for each iterated value of the field,
performing the steps of b) – g).

22. The processor executable medium as in claim 21, further comprising the steps of:

for each bitmap corresponding to the iterated value of the field;
determining if there is a matching bitmap previously created;
assigning a new index number as an index to a new entry that contains the bitmap
in the lookup table if no previous bitmap matches; and
assigning the index number of the previous bitmap to the bitmap if the bitmap
matches the previous bitmap.

23. The processor executable medium as in claim 22, further comprising the steps of:

004240-0845560

cross-producing a plurality of lookup tables corresponding to a plurality of fields
of the data header to form a next level lookup table.

24. The processor executable medium as in claim 23, wherein the step of cross-producing includes:
creating a next level bitmap from the cross-product of the plurality of bitmaps in the respective lookup tables;
determining if there is a matching next level bitmap previously created from the cross-product;
assigning a new next level index number as an index to a new entry that contains the next level bitmap in the next level lookup table if no previous next level bitmap matches; and
assigning the next level index number of the previous next level bitmap to the next level bitmap if the next level bitmap matches the previous next level bitmap.

25. The processor executable medium as in claim 24, further comprising the step of:
for N number of lookup tables, wherein N is an integer,
cross-producing the N number of lookup tables; and
repeating the cross-producing on resulting lookup tables until one final lookup table remains.

26. The process executable medium as in claim 25, wherein the final lookup table contains bitmaps, each bitmap representing a complete set of rules that match a data header value, the method further comprising the step of using the bitmap to find the first matching rule of the header value.

27. The process executable medium as in claim 25, wherein the final lookup table contains bitmaps, each bitmap representing a complete set of rules that match a data header value, the method further comprising the step of using the bitmap to find the best matching rule of the header value.

28. The process executable medium as in claim 25, wherein the final lookup table contains bitmaps, each bitmap representing a complete set of rules that match a data header value, the method further comprising the step of using the bitmap to find all the rules applicable to the header value.

29. An apparatus for performing data classification wherein data having a data header that is classified according to applicable rules, the apparatus comprising:
a memory having a lookup table, the lookup table having a plurality of indices associated with a plurality of bitmaps, each bitmap having a number of bits that indicates one or more matching rules; and
a circuit to receive the data header and use the data header as an entry to the lookup table to associate the data header with a bitmap in the lookup table.

30. The apparatus as in claim 29, wherein the memory is a flash memory.

31. The apparatus as in claim 29, wherein the memory is a content addressable memory (CAM).

32. The apparatus as in claim 29, wherein the circuit is a processor.

33. An apparatus for performing data classification wherein data having a data header that is classified according to applicable rules, the apparatus comprising:
a circuit to divide the data header into a plurality of fields;
at least two first level lookup tables stored in a memory to receive the fields of the data header, the first level lookup tables configured to generate first outputs responsive to the received fields; and
a final level lookup table stored in the memory to receive the first outputs from the first level lookup tables, the first outputs indexing a bitmap in the final level lookup table, the bitmap representing matching rules of the data header.

34. The apparatus as in claim 33, further comprising:
at least two intermediate level lookup tables stored in the memory to receive the first outputs from the first level lookup tables, the intermediate level lookup tables configured to generate intermediate outputs responsive to the received first outputs; and

the final level lookup table to receive the intermediate outputs from the intermediate level lookup tables, the intermediate outputs indexing the bitmap in the final level lookup table.

35. The apparatus as in claim 34, wherein the memory is a flash memory.

36. The apparatus as in claim 34, wherein the memory is a content addressable memory (CAM).

37. The apparatus as in claim 34, wherein the circuit is a processor.

38. An apparatus for performing data classification wherein data having a data header that is classified according to applicable rules, the apparatus comprising:

means for lookup having a plurality of indices associated with a plurality of bitmaps, each bitmap having a number of bits that indicates one or more matching rules; and
means for receiving and using the data header as an entry to the lookup means to associate the data header with a bitmap in the lookup table.

39. An apparatus for performing data classification wherein the data having a data header that is classified according to applicable rules, the apparatus comprising:

means for dividing the data header into a plurality of fields;
means for receiving the fields of the data header, field receiving means generating first outputs responsive to the received fields; and

- 6 means for receiving the first outputs from the field receiving means, the first outputs
- 7 indexing a bitmap in the first outputs receiving means, the bitmap representing matching
- 8 rules of the data header.

004240"0374560